What's New in Cocoa Touch

Session 104

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

What’s New in Cocoa Touch

Chris Parker
iPhone Frameworks Engineer

API

UIApplication

New methods

```
- (BOOL)canOpenURL:(NSURL *)url;
```

UIApplication

New delegate methods

```
- (BOOL)application:(UIApplication *)application
        didFinishLaunchingWithOptions:(NSDictionary *)opt;
```

```
NSString *const UIApplicationLaunchOptionsRemoteNotification
NSString *const UIApplicationLaunchOptionsURLKey
NSString *const UIApplicationLaunchOptionsSourceApplicationKey
```

UIApplication

New delegate methods

```
- (BOOL)application:(UIApplication *)application
      didFinishLaunchingWithOptions:(NSDictionary *)opt;
```

```
NSString *const UIApplicationLaunchOptionsRemoteNotification
NSString *const UIApplicationLaunchOptionsURLKey
NSString *const UIApplicationLaunchOptionsSourceApplicationKey
```

UIApplication

New delegate methods

```
- (BOOL)application:(UIApplication *)application
      didFinishLaunchingWithOptions:(NSDictionary *)opt;
```

```
NSString *const UIApplicationLaunchOptionsRemoteNotification
NSString *const UIApplicationLaunchOptionsURLKey
NSString *const UIApplicationLaunchOptionsSourceApplicationKey
```

UIApplication

Push notifications

```
-(void)registerForRemoteNotificationTypes:
                                (UIRemoteNotificationType)types;
-(void)unregisterForRemoteNotifications;
-(UIRemoteNotificationType)enabledRemoteNotificationTypes;
```

```
typedef enum {
    UIRemoteNotificationTypeNone    = 0,
    UIRemoteNotificationTypeBadge   = 1 << 0,
    UIRemoteNotificationTypeSound   = 1 << 1,
    UIRemoteNotificationTypeAlert   = 1 << 2
} UIRemoteNotificationType;
```

UIApplication

Push notifications

```
-(void)registerForRemoteNotificationTypes:
                                (UIRemoteNotificationType)types;
-(void)unregisterForRemoteNotifications;
-(UIRemoteNotificationType)enabledRemoteNotificationTypes;
```

```
typedef enum {
    UIRemoteNotificationTypeNone    = 0,
    UIRemoteNotificationTypeBadge   = 1 << 0,
    UIRemoteNotificationTypeSound   = 1 << 1,
    UIRemoteNotificationTypeAlert   = 1 << 2
} UIRemoteNotificationType;
```

UIApplication

Push notification delegate methods

```
-(void)application:(UIApplication *)app
       didRegisterForRemoteNotificationsWithDeviceToken:
                                          (NSData *)deviceToken;
-(void)application:(UIApplication *)app
       didFailToRegisterForRemoteNotificationsWithError:
                                          (NSError *)error;

-(void)application:(UIApplication *)app
       didReceiveRemoteNotification:(NSDictionary *)userInfo;
```

Apple Push Notification Service

Pacific Heights
Tuesday 5:00PM

UIDevice

Proximity monitoring

```
@property(nonatomic,getter=isProximityMonitoringEnabled)
                                        BOOL proximityMonitoringEnabled;

@property(nonatomic,readonly) BOOL proximityState;
```

```
NSString *const UIDeviceProximityStateDidChangeNotification;
```

UIDevice

Battery monitoring

```
@property(nonatomic,getter=isBatteryMonitoringEnabled)
                                        BOOL batteryMonitoringEnabled;
@property(nonatomic,readonly) UIDeviceBatteryState batteryState;
@property(nonatomic,readonly) float batteryLevel;
```

```
typedef enum {
    UIDeviceBatteryStateUnknown,
    UIDeviceBatteryStateUnplugged,
    UIDeviceBatteryStateCharging,
    UIDeviceBatteryStateFull
} UIDeviceBatteryState;
```

UIDevice

Battery monitoring

```
@property(nonatomic,getter=isBatteryMonitoringEnabled)
                                        BOOL batteryMonitoringEnabled;
@property(nonatomic,readonly) UIDeviceBatteryState batteryState;
@property(nonatomic,readonly) float batteryLevel;
```

```
typedef enum {
    UIDeviceBatteryStateUnknown,
    UIDeviceBatteryStateUnplugged,
    UIDeviceBatteryStateCharging,
    UIDeviceBatteryStateFull
} UIDeviceBatteryState;
```

UIDevice

Battery monitoring

```
NSString *const UIDeviceBatteryStateDidChangeNotification;
NSString *const UIDeviceBatteryLevelDidChangeNotification;
```

UIImageView

New methods and properties

```
- (id)initWithImage:(UIImage *)image
   highlightedImage:(UIImage *)highlightedImage;

@property(nonatomic,retain) UIImage *highlightedImage;
@property(nonatomic,getter=isHighlighted) BOOL highlighted;
@property(nonatomic,copy) NSArray *highlightedAnimationImages;
```

UINavigationController

New methods and properties

```
- (void)setViewControllers:(NSArray *)viewControllers
                  animated:(BOOL)animated;

@property(nonatomic,getter=isToolbarHidden) BOOL toolbarHidden;
- (void)setToolbarHidden:(BOOL)hidden
                  animated:(BOOL)animated;

@property(nonatomic,readonly) UIToolbar *toolbar;

@property(nonatomic, retain) NSArray *toolbarItems;
- (void)setToolbarItems:(NSArray *)toolbarItems
                  animated:(BOOL)animated
```

UIResponder

Motion detection

```
- (void)motionBegan:(UIEventSubtype)motion
          withEvent:(UIEvent *)event;

- (void)motionEnded:(UIEventSubtype)motion
          withEvent:(UIEvent *)event;

- (void)motionCancelled:(UIEventSubtype)motion
              withEvent:(UIEvent *)event;
```

UIEvent

Motion detection

```
@property (nonatomic,readonly) UIEventType type;
@property (nonatomic,readonly) UIEventSubtype subtype;

typedef enum {
    UIEventTypeTouches,
    UIEventTypeMotion,
} UIEventType;

typedef enum {
    UIEventSubtypeNone            = 0,
    UIEventSubtypeMotionShake = 1,
} UIEventSubtype;
```

UIResponder

Cut, copy, and paste

```
- (void)cut:(id)sender;
- (void)copy:(id)sender;
- (void)paste:(id)sender;

- (void)select:(id)sender;
- (void)selectAll:(id)sender;
```

- On UIApplication:

```
@property(nonatomic) BOOL applicationSupportsShakeToEdit;
```

UIResponder

New methods

```
- (BOOL)canPerformAction:(SEL)action
              withSender:(id)sender;
```

```
@property(nonatomic,readonly) NSUndoManager *undoManager;
```

NSUndoManager

Undo support

```
- (void)beginUndoGrouping;
- (void)endUndoGrouping;
- (NSInteger)groupingLevel;

- (void)undo;
- (void)redo;

- (BOOL)groupsByEvent;
- (void)setGroupsByEvent:(BOOL)groupsByEvent;
```

NSUndoManager

Undo support

```
- (void)beginUndoGrouping;
- (void)endUndoGrouping;
- (NSInteger)groupingLevel;

- (void)undo;
- (void)redo;

- (BOOL)groupsByEvent;
- (void)setGroupsByEvent:(BOOL)groupsByEvent;
```

NSUndoManager

Undo support

```
- (void)beginUndoGrouping;
- (void)endUndoGrouping;
- (NSInteger)groupingLevel;

- (void)undo;
- (void)redo;

- (BOOL)groupsByEvent;
- (void)setGroupsByEvent:(BOOL)groupsByEvent;
```

UIPasteboard

```
+ (UIPasteboard *)generalPasteboard;
+ (UIPasteboard *)pasteboardWithName:(NSString *)pasteboardName
                              create:(BOOL)create;
+ (UIPasteboard *)pasteboardWithUniqueName;
```

```
NSString *const UIPasteboardChangedNotification;
NSString *const    UIPasteboardChangedTypesAddedKey;
NSString *const    UIPasteboardChangedTypesRemovedKey;

NSString *const UIPasteboardRemovedNotification;
```

UIPasteboard

- Universal Type Identifiers
- Single and multiple items
- Data extensions
  - Strings
  - URLs
  - Images
  - Colors

UIMenuController

Showing the pasteboard interface on your views

```
+ (UIMenuController *)sharedMenuController;
```

```
@property(nonatomic,getter=isMenuVisible) BOOL menuVisible;
- (void)setMenuVisible:(BOOL)menuVisible animated:(BOOL)animated;

- (void)setTargetRect:(CGRect)targetRect inView:(UIView *)view;

- (void)update;
```

UIMenuController

Showing the pasteboard interface on your views

```
+ (UIMenuController *)sharedMenuController;
```

```
@property(nonatomic,getter=isMenuVisible) BOOL menuVisible;
- (void)setMenuVisible:(BOOL)menuVisible animated:(BOOL)animated;

- (void)setTargetRect:(CGRect)targetRect inView:(UIView *)view;

- (void)update;
```

Cut, Copy, and Paste on iPhone

Russian Hill
Wednesday 3:30PM

UIScrollView

New methods and properties

```
const float UIScrollViewDecelerationRateNormal;
const float UIScrollViewDecelerationRateFast;

@property(nonatomic) float decelerationRate;
```

```
@property(nonatomic) float zoomScale;

- (void)setZoomScale:(float)scale
            animated:(BOOL)animated;

- (void)zoomToRect:(CGRect)rect
          animated:(BOOL)animated;
```

UITableView

Section index

```
NSString *const UITableViewIndexSearch;
```

```
- (void)reloadSectionIndexTitles;
```

UITableView

New methods

```
-(void)reloadSections:(NSIndexSet *)sections
     withRowAnimation:(UITableViewRowAnimation)animation;

-(void)reloadRowsAtIndexPaths:(NSArray *)indexPaths
             withRowAnimation:(UITableViewRowAnimation)animation;
```

```
@property(nonatomic) BOOL allowsSelection;
```

UITableView

New delegate methods

```
-(NSIndexPath *)tableView:(UITableView *)tableView
                willDeselectRowAtIndexPath:(NSIndexPath *)indexPath;

-(void)tableView:(UITableView *)tableView
       didDeselectRowAtIndexPath:(NSIndexPath *)indexPath;
```

UITableView

New delegate methods

```
- (NSString *)tableView:(UITableView *)tableView
           titleForDeleteConfirmationButtonForRowAtIndexPath:
                                                 (NSIndexPath *)indexPath;
```

UITableViewCell

New methods

```
- (id)initWithStyle:(UITableViewCellStyle)style
    reuseIdentifier:(NSString *)reuseIdentifier;
```

```
typedef enum {
    UITableViewCellStyleDefault,
    UITableViewCellStyleValue1,
    UITableViewCellStyleValue2,
    UITableViewCellStyleSubtitle
} UITableViewCellStyle;
```

UITableViewCell

New methods

```
- (id)initWithStyle:(UITableViewCellStyle)style
    reuseIdentifier:(NSString *)reuseIdentifier;
```

```
typedef enum {
    UITableViewCellStyleDefault,
    UITableViewCellStyleValue1,
    UITableViewCellStyleValue2,
    UITableViewCellStyleSubtitle
} UITableViewCellStyle;
```

UITableViewCell

Styles

UITableViewCellStyleDefault

UITableViewCellStyleValue1

UITableViewCellStyleValue2

UITableViewCellStyleSubtitle

UITableViewCell

New methods and properties

```
@property(nonatomic,readonly,retain) UIImageView *imageView;
@property(nonatomic,readonly,retain) UILabel *textLabel;
@property(nonatomic,readonly,retain) UILabel *detailTextLabel;
```

```
- (void)willTransitionToState:(UITableViewCellStateMask)state;
- (void)didTransitionToState:(UITableViewCellStateMask)state;
```

```
enum {
    UITableViewCellStateDefaultMask                 = 0,
    UITableViewCellStateShowingEditControlMask      = 1 << 0,
    UITableViewCellStateShowingDeleteConfirmationMask   = 1 << 1
};
typedef NSUInteger UITableViewCellStateMask;
```

UITableViewCell

New methods and properties

```
@property(nonatomic,readonly,retain) UIImageView *imageView;
@property(nonatomic,readonly,retain) UILabel *textLabel;
@property(nonatomic,readonly,retain) UILabel *detailTextLabel;
```

```
- (void)willTransitionToState:(UITableViewCellStateMask)state;
- (void)didTransitionToState:(UITableViewCellStateMask)state;
```

```
enum {
    UITableViewCellStateDefaultMask                 = 0,
    UITableViewCellStateShowingEditControlMask      = 1 << 0,
    UITableViewCellStateShowingDeleteConfirmationMask   = 1 << 1
};
typedef NSUInteger UITableViewCellStateMask;
```

UILocalizedIndexedCollation

Localized sorting of index

```
+ (id)currentCollation;

@property(nonatomic, readonly) NSArray *sectionTitles;
@property(nonatomic, readonly) NSArray *sectionIndexTitles;

- (NSInteger)sectionForSectionIndexTitleAtIndex:(NSInteger)indexTitleIndex;

- (NSInteger)sectionForObject:(id)object
      collationStringSelector:(SEL)selector;

- (NSArray *)sortedArrayFromArray:(NSArray *)array
          collationStringSelector:(SEL)selector;
```

UILocalizedIndexedCollation

Localized sorting of index

```
+ (id)currentCollation;

@property(nonatomic, readonly) NSArray *sectionTitles;
@property(nonatomic, readonly) NSArray *sectionIndexTitles;

- (NSInteger)sectionForSectionIndexTitleAtIndex:(NSInteger)indexTitleIndex;

- (NSInteger)sectionForObject:(id)object
      collationStringSelector:(SEL)selector;

- (NSArray *)sortedArrayFromArray:(NSArray *)array
          collationStringSelector:(SEL)selector;
```

UILocalizedIndexedCollation

Localized sorting of index

```
+ (id)currentCollation;

@property(nonatomic, readonly) NSArray *sectionTitles;
@property(nonatomic, readonly) NSArray *sectionIndexTitles;

- (NSInteger)sectionForSectionIndexTitleAtIndex:(NSInteger)indexTitleIndex;

- (NSInteger)sectionForObject:(id)object
      collationStringSelector:(SEL)selector;

- (NSArray *)sortedArrayFromArray:(NSArray *)array
          collationStringSelector:(SEL)selector;
```

UILocalizedIndexedCollation

Localized sorting of index

```
+ (id)currentCollation;

@property(nonatomic, readonly) NSArray *sectionTitles;
@property(nonatomic, readonly) NSArray *sectionIndexTitles;

- (NSInteger)sectionForSectionIndexTitleAtIndex:(NSInteger)indexTitleIndex;

- (NSInteger)sectionForObject:(id)object
      collationStringSelector:(SEL)selector;

- (NSArray *)sortedArrayFromArray:(NSArray *)array
          collationStringSelector:(SEL)selector;
```

UISearchBar

Scopes

UISearchBar

Scopes

```
@property(nonatomic) BOOL showsScopeBar;
@property(nonatomic,copy) NSArray *scopeButtonTitles;
@property(nonatomic) NSInteger selectedScopeButtonIndex;
```

UISearchBar

Scopes

```
@property(nonatomic) BOOL showsScopeBar;
@property(nonatomic,copy) NSArray *scopeButtonTitles;
@property(nonatomic) NSInteger selectedScopeButtonIndex;
```

```
- (void)searchBar:(UISearchBar *)searchBar
selectedScopeButtonIndexDidChange:(NSInteger)selectedScope;
```

UISearchDisplayController

Mediating search controller

```
- (id)initWithSearchBar:(UISearchBar *)searchBar
     contentsController:(UIViewController *)viewController;
```

```
@property(nonatomic,assign) id <UISearchDisplayDelegate> delegate;
@property(nonatomic,getter=isActive) BOOL active;
- (void)setActive:(BOOL)visible animated:(BOOL)animated;

@property(nonatomic,readonly) UISearchBar *searchBar;
@property(nonatomic,readonly) UIViewController *searchContentsController;
@property(nonatomic,readonly) UITableView *searchResultsTableView;
@property(nonatomic,assign) id <UITableViewDataSource>
                                                *searchResultsDataSource;
@property(nonatomic,assign) id <UITableViewDelegate>
                                                    searchResultsDelegate;
```

UISearchDisplayController

Mediating search controller

```
- (id)initWithSearchBar:(UISearchBar *)searchBar
     contentsController:(UIViewController *)viewController;
```

```
@property(nonatomic,assign) id <UISearchDisplayDelegate> delegate;
@property(nonatomic,getter=isActive) BOOL active;
- (void)setActive:(BOOL)visible animated:(BOOL)animated;

@property(nonatomic,readonly) UISearchBar *searchBar;
@property(nonatomic,readonly) UIViewController *searchContentsController;
@property(nonatomic,readonly) UITableView *searchResultsTableView;
@property(nonatomic,assign) id <UITableViewDataSource>
                                                *searchResultsDataSource;
@property(nonatomic,assign) id <UITableViewDelegate>
                                                  *searchResultsDelegate;
```

UISearchDisplayController

Mediating search controller

```
- (void)searchDisplayControllerWillBeginSearch:(UISearchDisplayController *)controller;
- (void)searchDisplayControllerDidBeginSearch:(UISearchDisplayController *)controller;
- (void)searchDisplayControllerWillEndSearch:(UISearchDisplayController *)controller;
- (void)searchDisplayControllerDidEndSearch:(UISearchDisplayController *)controller;

 - (void)searchDisplayController:(UISearchDisplayController *)controller
   didLoadSearchResultsTableView:(UITableView *)tableView;

 - (void)searchDisplayController:(UISearchDisplayController *)controller
willUnloadSearchResultsTableView:(UITableView *)tableView;
```

UISearchDisplayController

Mediating search controller

```
- (void)searchDisplayControllerWillBeginSearch:(UISearchDisplayController *)controller;
- (void)searchDisplayControllerDidBeginSearch:(UISearchDisplayController *)controller;
- (void)searchDisplayControllerWillEndSearch:(UISearchDisplayController *)controller;
- (void)searchDisplayControllerDidEndSearch:(UISearchDisplayController *)controller;

 - (void)searchDisplayController:(UISearchDisplayController *)controller
   didLoadSearchResultsTableView:(UITableView *)tableView;

 - (void)searchDisplayController:(UISearchDisplayController *)controller
willUnloadSearchResultsTableView:(UITableView *)tableView;
```

UISearchDisplayController

Mediating search controller

```
- (void)searchDisplayController:(UISearchDisplayController *)controller
 willShowSearchResultsTableView:(UITableView *)tableView;
- (void)searchDisplayController:(UISearchDisplayController *)controller
  didShowSearchResultsTableView:(UITableView *)tableView;
- (void)searchDisplayController:(UISearchDisplayController *)controller
 willHideSearchResultsTableView:(UITableView *)tableView;
- (void)searchDisplayController:(UISearchDisplayController *)controller
  didHideSearchResultsTableView:(UITableView *)tableView;

 - (BOOL)searchDisplayController:(UISearchDisplayController *)controller
shouldReloadTableForSearchString:(NSString *)searchString;
 - (BOOL)searchDisplayController:(UISearchDisplayController *)controller
 shouldReloadTableForSearchScope:(NSInteger)searchOption;
```

UISearchDisplayController

Mediating search controller

```
- (void)searchDisplayController:(UISearchDisplayController *)controller
 willShowSearchResultsTableView:(UITableView *)tableView;
- (void)searchDisplayController:(UISearchDisplayController *)controller
  didShowSearchResultsTableView:(UITableView *)tableView;
- (void)searchDisplayController:(UISearchDisplayController *)controller
 willHideSearchResultsTableView:(UITableView *)tableView;
- (void)searchDisplayController:(UISearchDisplayController *)controller
  didHideSearchResultsTableView:(UITableView *)tableView;

 - (BOOL)searchDisplayController:(UISearchDisplayController *)controller
shouldReloadTableForSearchString:(NSString *)searchString;
 - (BOOL)searchDisplayController:(UISearchDisplayController *)controller
 shouldReloadTableForSearchScope:(NSInteger)searchOption;
```

UITabBarController

New methods

```
@property(nonatomic,readonly) UITabBar *tabBar;
```

```
  - (BOOL)tabBarController:(UITabBarController *)tbc
shouldSelectViewController:(UIViewController *)vc;

            - (void)tabBarController:(UITabBarController *)tbc
willBeginCustomizingViewControllers:(NSArray *)vcs;

          - (void)tabBarController:(UITabBarController *)tbc
willEndCustomizingViewControllers:(NSArray *)vcs
                          changed:(BOOL)changed;

        - (void)tabBarController:(UITabBarController *)tbc
didEndCustomizingViewControllers:(NSArray *)vcs
                          changed:(BOOL)changed;
```

UITabBarController

New methods

```
@property(nonatomic,readonly) UITabBar *tabBar;
```

```
  - (BOOL)tabBarController:(UITabBarController *)tbc
shouldSelectViewController:(UIViewController *)vc;

            - (void)tabBarController:(UITabBarController *)tbc
willBeginCustomizingViewControllers:(NSArray *)vcs;

          - (void)tabBarController:(UITabBarController *)tbc
willEndCustomizingViewControllers:(NSArray *)vcs
                          changed:(BOOL)changed;

        - (void)tabBarController:(UITabBarController *)tbc
didEndCustomizingViewControllers:(NSArray *)vcs
                          changed:(BOOL)changed;
```

UIViewController

New methods

```
- (void)viewDidUnload;
- (BOOL)isViewLoaded;
```

```
- (void)willAnimateRotationToInterfaceOrientation:
                                (UIInterfaceOrientation)toInterfaceOrientation
        duration:(NSTimeInterval)duration;
```

UIViewController

New methods

```
- (void)viewDidUnload;
- (BOOL)isViewLoaded;
```

```
- (void)willAnimateRotationToInterfaceOrientation:
                                (UIInterfaceOrientation)toInterfaceOrientation
        duration:(NSTimeInterval)duration;
```

UIViewController

New methods

```
@property(nonatomic,assign) BOOL wantsFullScreenLayout;
```

UIViewControllers

Modal transitions

```
@property(nonatomic,assign) UIModalTransitionStyle modalTransitionStyle;

typedef enum {
    UIModalTransitionStyleCoverVertical = 0,
    UIModalTransitionStyleFlipHorizontal,
    UIModalTransitionStyleCrossDissolve
} UIModalTransitionStyle;
```

UIViewControllers

Modal transitions

```
@property(nonatomic,assign) UIModalTransitionStyle modalTransitionStyle;

typedef enum {
    UIModalTransitionStyleCoverVertical = 0,
    UIModalTransitionStyleFlipHorizontal,
    UIModalTransitionStyleCrossDissolve
} UIModalTransitionStyle;
```

iPhone View Controller Techniques

Presidio
Thursday 5:00PM

New Features

`

Info plist Keys

- Group opacity

```
<key>UIViewGroupOpacity</key>
<true>
```

- Edge anti-aliasing

```
<key>UIViewEdgeAntialiasing</key>
<true>
```

Demo

Data Detectors

UITextView and UIWebView

```
enum {
    UIDataDetectorTypePhoneNumber    = 1 << 0,
    UIDataDetectorTypeLink           = 1 << 1,

    UIDataDetectorTypeNone           = 0,
    UIDataDetectorTypeAll            = NSUIntegerMax
};
typedef NSUInteger UIDataDetectorTypes;
```

Data Detectors

UITextView and UIWebView

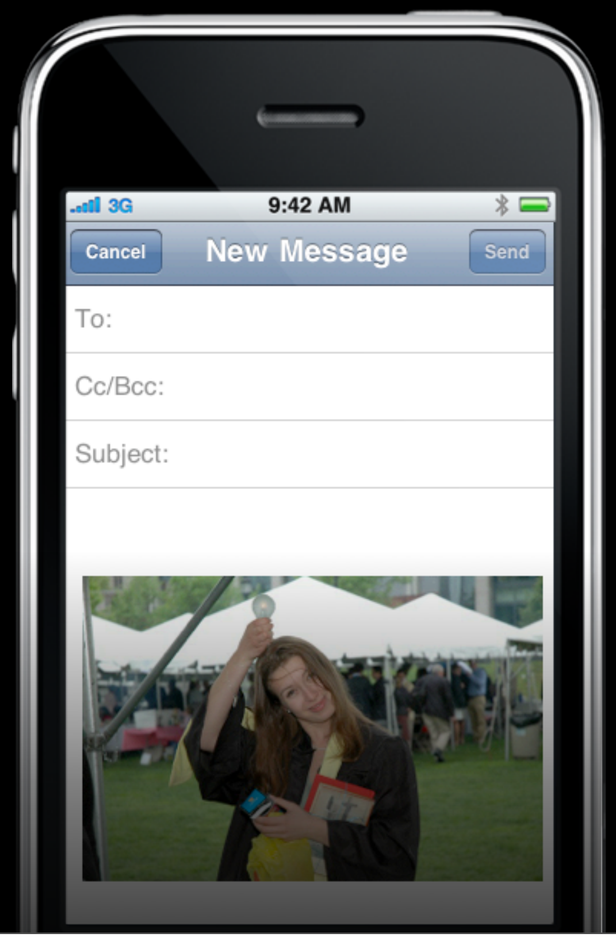
```
enum {
    UIDataDetectorTypePhoneNumber   = 1 << 0,
    UIDataDetectorTypeLink          = 1 << 1,

    UIDataDetectorTypeNone          = 0,
    UIDataDetectorTypeAll           = NSUIntegerMax
};
typedef NSUInteger UIDataDetectorTypes;
```

```
@property(nonatomic) UIDataDetectorTypes dataDetectorTypes;
```

Message Sending

In-application email

```
#import <MessageUI/MessageUI.h>
@interface MFMailComposeViewController : UINavigationController
```



Message Sending

In-application email

```
+ (BOOL)canSendMail;

- (void)setSubject:(NSString *)subject;
- (void)setToRecipients:(NSArray *)toRecipients;
- (void)setCcRecipients:(NSArray *)ccRecipients;
- (void)setBccRecipients:(NSArray *)bccRecipients;
- (void)setMessageBody:(NSString *)body
                isHTML:(BOOL)isHTML;
- (void)addAttachmentData:(NSData *)attachment
                 mimeType:(NSString *)mimeType
                 fileName:(NSString *)filename;
```

Message Sending

In-application email

```
+ (BOOL)canSendMail;

- (void)setSubject:(NSString *)subject;
- (void)setToRecipients:(NSArray *)toRecipients;
- (void)setCcRecipients:(NSArray *)ccRecipients;
- (void)setBccRecipients:(NSArray *)bccRecipients;
- (void)setMessageBody:(NSString *)body
                isHTML:(BOOL)isHTML;
- (void)addAttachmentData:(NSData *)attachment
                 mimeType:(NSString *)mimeType
                 fileName:(NSString *)filename;
```

Message Sending

In-application email

```
- (void)mailComposeController:(MFMailComposeViewController *)controller
          didFinishWithResult:(MFMailComposeResult)result
                        error:(NSError *)error;
```

Message Sending

In-application email

```
- (void)mailComposeController:(MFMailComposeViewController *)controller
          didFinishWithResult:(MFMailComposeResult)result
                        error:(NSError *)error;
```

```
enum MFMailComposeResult {
    MFMailComposeResultCancelled,
    MFMailComposeResultSaved,
    MFMailComposeResultSent,
    MFMailComposeResultFailed
};
typedef enum MFMailComposeResult MFMailComposeResult;
```

Message Sending

In-application email

```
- (void)mailComposeController:(MFMailComposeViewController *)controller
          didFinishWithResult:(MFMailComposeResult)result
                        error:(NSError *)error;
```

```
NSString *const MFMailComposeErrorDomain;

enum MFMailComposeErrorCode {
    MFMailComposeErrorCodeSaveFailed,
    MFMailComposeErrorCodeSendFailed
};
typedef enum MFMailComposeErrorCode MFMailComposeErrorCode;
```

StoreKit

In-application purchases

- Products
- Payments
- Payment queue management

In-App Purchase on iPhone

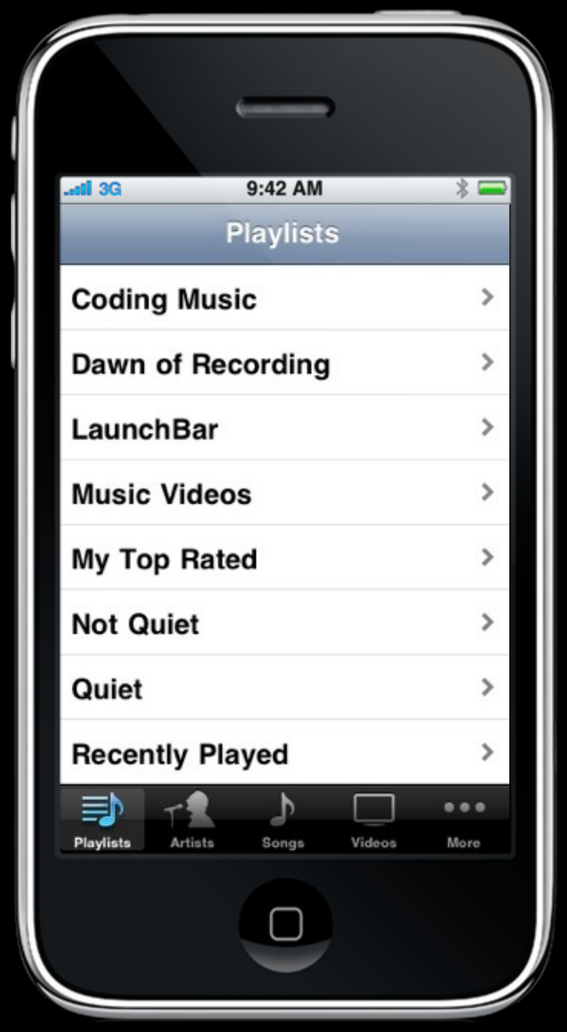
Presidio
Wednesday 10:30AM

GameKit

Networking and discovery

- Peer-to-peer networking
- Discovering and selecting peers
- Communicating application data and game state
  - Over Bluetooth
  - Without pairing!
- In-game voice chat

Peer-to-Peer Networking with Game Kit	Presidio Wednesday 9:00AM



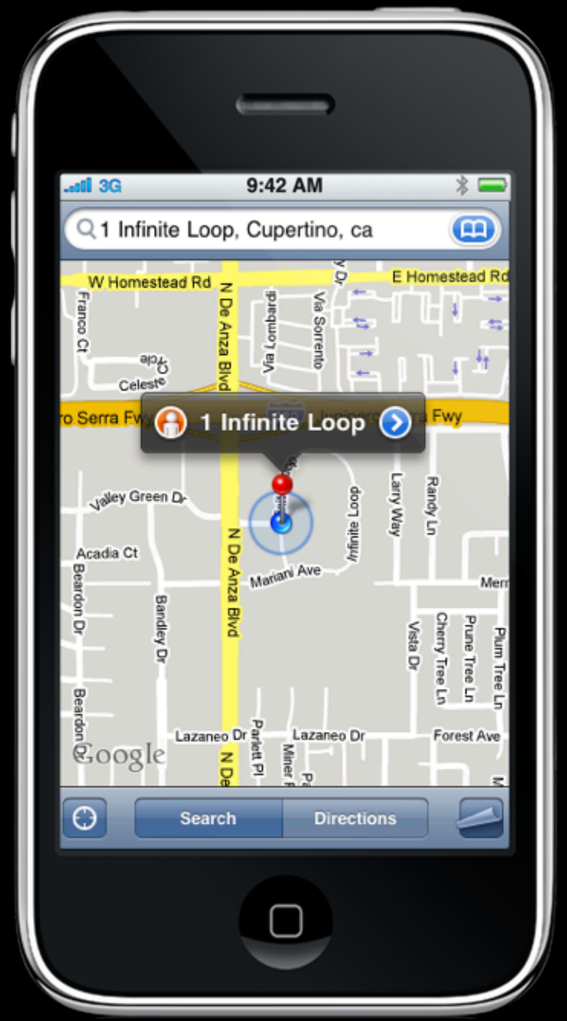
MediaPlayer

New access features

- Songs
- Podcasts
- Audiobooks
- Custom searches
- Playlists

Accessing the iPod Library

Pacific Heights
Thursday 5:00PM



MapKit

Embedded map views

- Annotations
- Pins
- Custom markers
- Reverse geocoding

Embedding Maps in iPhone Applications	Presidio Thursday 10:30AM

CoreLocation

You are here

- Compass
  - Heading
  - Update notifications

Finding Your Place with Core Location	Presidio Thursday 2:00PM

Accessibility

VoiceOver

- UIAccessibility protocol
- All standard controls
- Additional annotations

iPhone Accessibility: Make Your App Compatible with VoiceOver

North Beach
Thursday 9:00AM

Core Data

Object persistence management

- Entities
- Relationships
- Persistence
- Interoperability

Introduction to Core Data on iPhone	Marina Tuesday 5:00PM
What's New in Core Data	North Beach Wednesday 10:30AM

Related Sessions

Perfecting Your iPhone Table Views	Pacific Heights Tuesday 2:00PM
iPhone Views and Animations	Pacific Heights Tuesday 3:30PM
Mastering iPhone Scroll Views	Pacific Heights Wednesday 2:00PM
Cut, Copy, and Paste on iPhone	Russian Hill Wednesday 3:30PM
Processing Multi-Touch Events on iPhone	Pacific Heights Thursday 9:00AM
iPhone View Controller Techniques	Presidio Thursday 5:00PM

Labs

- Address Book
- Push Notifications
- Audio
- Calendar and Contacts
- Core Animation
- Core Data (Mac and iPhone)
- Core Location and MapKit
- Accessories
- Graphics and Media
- In-app Purchase
- Performance
- Security
- Table and Scroll Views
- View Controllers
- iPod Library Access
- Migrating SQLite to Core Data
- Game Development

More Information

Matt Drance
iPhone and Mac Application Frameworks Evangelist
mdrance@apple.com

Documentation
iPhone Dev Center
http://developer.apple.com/iphone/

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09